A Low Power and Reliable Charge Pump Design for Phase Change Memories

Lei Jiang† Bo Zhao† Jun Yang† Youtao Zhang‡
† Electrical and Computer Engineering Department ‡ Computer Science Department
University of Pittsburgh, Pittsburgh
†{lej16,boz6,juy9}@pitt.edu ‡zhangyt@cs.pitt.edu

Abstract

The emerging Phase Change Memory (PCM) technology exhibits excellent scalability and density potentials. At the same time, they require high current and high voltages to switch cell states. Their working voltages are provided by CMOS-compatible on-chip charge pumps (CPs). Unfortunately, CPs and particularly those for RESET, have a large parasitic power (a dominant component in total power loss) during operations, which significantly degrades their energy efficiency. In addition, CPs seriously suffer from the Time-Dependent Dielectric Breakdown (TDDB) problem due to their boosted operation voltage. To maintain a reasonable lifetime of CPs, existing solutions actively switch them on per-operation basis, resulting in large performance degradation.

In this paper, we address the above issues through two designs — Reset_Sch (RESET scheduling) and CP_Sch (CP scheduling). Reset_Sch schedules when to perform a RESET for different cells upon writing a PCM line. It significantly reduces the power loss, and peak working power of RESET CP. CP_Sch incorporates a fast READ CP design to provide fast charge-up time for reads and minimize performance penalty. Our experimental results show that on average, 70% of power loss for RESET CP can be reduced; and performance loss can be reduced from 16% to 2% while achieving a 16% improvement in reliability.

1. Introduction

Phase Change Memory (PCM), a promising non-volatile memory, is projected to replace a substantial portion of traditional DRAM in future memory hierarchies [20, 32, 38]. PCM has excellent scalability, zero cell leakage, and fast read speed close to that of DRAM. The limitations, however, include much longer write latency and higher write power than DRAM writes. The current required to program a PCM cell is orders of magnitude higher than that for a DRAM cell. High power requirement has created serious challenges in power delivery as well as write concurrency. A flurry of prior work has been proposed to suppress or manage the write power [5, 11, 15, 38],

A parallel effort to power management is on increasing PCM device density. High density can be achieved through using a smaller access device, such as bipolar junction transistor (BJT) and diode, as opposed to a MOS transistor, to supply current into PCM cell. Recent PCM innovations use a diode as the access device to achieve a minimum of $4F^2$ cell size as compared to $7F^2$ DRAM cell size [6, 21]. Such high density can be further multiplied by exploiting the large resistance distance between fully crystalline and fully amorphous states of PCM, forming multi-level cells (MLC) which reduce cost per bit by storing more than one bit per cell.

High power consumption has become a major challenge in designing PCM based memory systems [11, 15]. The working voltage needs to be boosted from 1.8V (V_{dd}) to 2.8V, 3.0V or even 5.0V for BJT-, MOS- and diode-switched PCM respectively [1, 17, 21]. Those high voltages are provided by different types of CMOS-compatible on-chip charge pumps (CPs) [26], which convert a lower input voltage to higher output voltages. There are major limitations to CPs in PCM chips. First, a CP typically consists of cascaded stages of large capacitors and wide transistors. Each stage elevates the voltage by a certain amount. Charging and discharging consume large *parasitic* power due to parasitic capacitance proportional to those large capacitors [26, 35]. In addition, the leakage power of CPs is usually quite large as a result of the wide, strong transistors and high voltages on internal nodes and the output [35]. Also, CPs dissipate significant power on its own peripheral circuits such as controls, drivers, clock generation and distribution. The parasitic, leakage and peripheral circuit power are significant sources of power loss of CP. We term it `wasted power` in this work. This is also why the power conversion efficiency of CP, defined as the ratio between output and input power, is usually very low. As low as 20% of efficiency has been reported for a CP with current load in several PCM chips[1] [21]. To supply enough output current, either larger input current of a single CP is needed, or more CP units are necessary. As a result, CPs consume large chip area, e.g., ∼20% [21], as well. Our evaluation shows that the total power dissipated by the CPs accounts for more than 81% of the total memory power, where 60% is due to just the parasitic power. Hence, it becomes increasingly important to design effective schemes to reduce power loss of CPs.

The second limitation of CP is its impact on performance due to its long charge latency. Due to the huge load capacitance on the power supply network, charging a CP to a target output voltage takes excessive amount of time, e.g., 200∼300ns [17, 21]. Recent PCM chip demonstrations [6, 21, 17] discharge CPs forcefully at the end of each memory access and charge them up again at the beginning of each request.

[1] A CP with capacitive load has much higher efficiency, which will be discussed in Section 3.1.

978-1-4799-4394-4/14/$31.00

The main reason for frequent switching of CP is to maintain its reliability [17], despite the significant performance and energy loss. This is because as technology scales, the thickness of the gate oxide becomes smaller and hence, the operation voltage of transistors must be lowered to mitigate the *time-dependent gate oxide breakdown* (TDDB) [13]. However, CPs are stressed under higher-than V_{dd} voltages while working, and thus, are more vulnerable to TDDB. Hence, switching off CPs after each operation can greatly reduce the time it is stressed, as memory chip idle time predominates over busy time. Our evaluation will show that an average of 16% of performance degradation with 0.5% of energy increase can be observed from such a per-operation switching scheme, indicating a significant trade-off between performance, energy and reliability.

In this paper, we propose techniques to tackle the aforementioned main limitations to PCM CPs. Our contributions are as follows.

- We propose *RESET scheduling*, a scheme that significantly reduces the demand for large-sized RESET CPs. This is achieved through reducing the peak power in writing a memory line via scheduling the high-power RESET operations over the entire duration of the write, without prolonging the write latency. Such scheduling effectively diminishes the RESET CP area and wasted power by 70%.
- We found that frequent switching of CPs on per-operation basis has little impact on energy consumption because the charge and discharge energy is offset by the leakage energy saving during off time of the CP. However, significant performance loss (16%) is observed due to the charge-up latency upon arrival of new requests, especially the reads which are on critical path. We therefore propose a fast charging scheme for read CPs, which is 4× faster than the naïve charging scheme.
- We provide detailed CP modeling, and simulated our proposed techniques on MOS-, BJT-, diode-switched PCMs. We also tested both SLC and MLC structures. The overall reduction in wasted power are observed to be between 37% and 49% for different access devices or cell designs. These results prove that the proposed techniques are effective and generally applicable to different PCM designs.

2. Background

2.1. High Density PCM

High density PCM can be achieved by using a smaller access device (Figure 2(a)) with a compact structure because the area of the access device dominates the size of a cell. Three types of access devices have been implemented in PCM prototypes: MOS [17], bipolar junction transistor (BJT) [1] and diode [21]. They, in that order, create decreasing PCM cell sizes, as shown in Figure 1(a). As a result, a PCM bank built with MOS access device will be larger than with BJT. And a diode-switched PCM array will be the smallest. A diode-switched PCM achieves a minimum of $4F^2$ cell size. The cell has a vertical structure including a bit-line, a top electrode contact, a phase-change material such as GST, a self-aligned bottom electrode contact and a diode [21]. We modeled a 1GB PCM following a prototype [6] with those three different access devices, and measured their area using NVsim [8]. The results, shown in Figure 1(b), are all normalized to the area of the MOS-switched PCM array. As we can see, the BJT-switched PCM array is only 63% of the MOS-switched array in area. The diode-switched array further reduces it to 44%. A multi-level cell (MLC) with a diode (MLC-Dio) achieves the minimum area: only 26% of the original area is required. Due to such great density advantage, many recent PCM prototypes and products adopt diode-switched PCM cell design in 58*nm* [7] (2011) and 20*nm* [6] (2012).

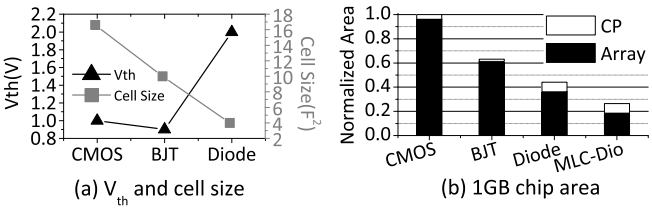

Figure 1: Trade-off between V_{th} and chip density.

However, a diode has the highest V_{th} among the three access devices [21], as also shown in Figure 1(a). Additionally, diode-switched PCM has a larger parasitic resistance on its bit-line in high-density PCM array architecture [6]. Hence, it needs higher read and write voltages than MOS- and BJT-switched PCM, which means that larger CPs for reads and writes are necessary to overcome the V_{th} and parasitic resistance. The area of the corresponding CP for three different devices are also shown in Figure 1(b). Regardless of the reduction of the entire chip area, the area of CP enlarges more than two times from MOS-, to diode-switched arrays. CP occupies 43% of the array area in MLC-Dio. As will be introduced later, on-chip CP has low conversion efficiency (only ∼20%). When the area proportion of CP grows, more leakage power is dissipated and higher power attrition appears. This is the problem we will address in this work.

2.2. Multi-level Cell (MLC) PCM

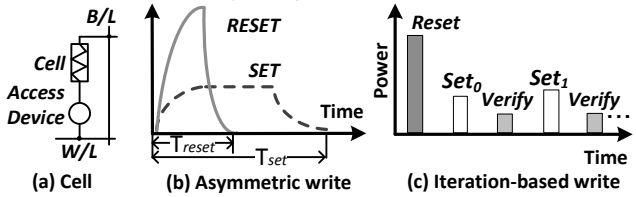

Figure 2: PCM basics.

The phase change material used in PCM, such as GST, has two stable states — fully crystalline and fully amorphous states. The resistance difference between these two states is often more than $10^4\times$ apart [33]. Multi-level cells (MLC) can thus be implemented through creating multiple intermediate states to represent more binary bits, e.g. four states for two bits. The intermediate states are partially crystalline and partially amorphous material states.

There are two PCM write operations — RESET and SET. The RESET is performed by applying a large but short pulse to the GST material and converts it from crystalline to amorphous state. The SET is performed by applying a smaller but longer pulse for the reverse state transition, as illustrated in Figure 2(b). Such an asymmetric write characteristic indicates that the PCM power consumption is largely determined by the RESET operation. Its high pulse also requires a significantly larger CP than the SET and read operations do.

Programming an MLC requires multiple steps to reach a target resistance level due to the significant write nondeterminism caused by process variations and material composition fluctuation [2, 24]. Also, a cell value is represented by a resistance range rather than a specific point. An iteration-based write scheme is adopted to program a cell into a target resistance range, as illustrated in Figure 2(c). This is the widely used Program-and-Verify (P&V) scheme [25, 27] that first applies a RESET pulse to place cells that need to be changed into similar initial states, and then applies a number of SET pulses, verified by reads in each iteration, to ensure the write accuracy.

Recent studies revealed that writing different cell values requires different numbers of iterations [30, 16]. In 2-bit MLC, for example, writing 00 can finish immediately after the RESET iteration. Writing 01 or 10 requires more iterations than writing 00 or 11. It is this property that we will leverage to develop a scheduling scheme to lower the peak power during a write. Also, based on the cell values to be written, a PCM line write may require different numbers of iterations at different times. We will adopt the same scheme in this paper. Detailed parameters are reported in Section 5.

2.3. The Baseline Memory Architecture

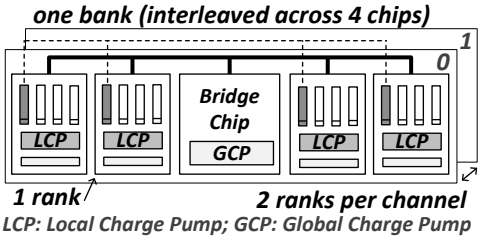

Figure 3: The baseline MLC PCM-based memory system.

The baseline memory architecture (as shown in Figure 3) is derived from a previous MLC PCM model [15] and a new memory interface LPDDR2-NVM (Low Power DDR2 for Non-Volatile Memory) [6]. Previous work [15] adopted eight 8-bit PCM chips to constitute eight banks on one DDR2 DIMM. LPDDR2-NVM interface only supports 8, 16 and 32 bits wide chips. In this paper, we set chip width as 16-bit. Hence, there are now four chips per rank, and each logic bank spreads across four chips. The channel adopts the dual line package (DLP) architecture [22], which has single LPDDR2-NVM package on both sides of the board. One LPDDR2-NVM package has four 1GB chips, as with the prototype demonstrated in [6]. It was also demonstrated that one chip can support 128 parallel RESETs independently. In this paper, we assume one MLC PCM chip can support 140 concurrent writes, similar to that in [15]. We will prove that we can reduce this need by 70%.

Due to non-deterministic MLC PCM write, a bridge chip is integrated on-DIMM to regulate the P&V iteration levels [9]. Regulating MLC write by the bridge chip instead of the memory controller minimizes sub-optimal utilization of the memory bus. Finally, a large DRAM cache for the PCM memory is assumed, which can buffer write-intensive lines to benefit PCM in both endurance and power.

We also adopt two levels of CP design as proposed in [15]: local CP (LCP) and global CP (GCP). The latter was introduced to distribute power according to the need of each individual chip. In summary, we adopt a state-of-the-art baseline where CPs are designed to ensure that the available power is efficiently used.

3. Charge Pump Basics and Modeling

Recent advances in PCM incorporate both on-chip and off-chip power supplies [6, 17, 21, 23]. Even though external power supply was available, on-chip CPs still predominate, as demonstrated in [1, 6, 7, 12, 17, 21]. This is because: 1) PCM requires multiple boosted voltages in different components of a chip, and hence, single external power rail is insufficient to achieve that. 2) Memory chips only have tens of pins, in contrast to hundreds of them in microprocessors [14]. Adding more pins for multiple external power rails directly increases the cost of the chip and reduces the already thin profit margin for memory chips. 3) The write voltages need a fast and fine-grained control (for pulse shaping, location compensation [37], etc.) which cannot be achieved via external power. And 4) fully depending on external power supply is not a portable solution because different vendors may require different voltage boost levels. Hence, we focus on optimizing on-chip CMOS-compatible CPs and present first the models used in this paper.

3.1. CMOS-compatible On-chip Charge Pumps

A CP converts the supply voltage V_{dd} to a DC output voltage V_{out} higher than V_{dd}. CMOS-compatible CP consists only of capacitors and switches, so it can be integrated on-chip easily. A CP has several stages, each of which elevates the voltage a little, as shown in Figure 4(a). Adding more stages can raise the output voltage to a target level that is multiple times higher than V_{dd} [26].

There are two types of CPs [26]: one with purely capacitive load and the other with a current load. The former does not need to supply any current and only provides a target output voltage, which can be applied on X/Y decoders to reduce the parasitic resistances of the transistor switches along the read/write current path [17, 6]. It has a negligible area overhead and almost perfect power conversion efficiency, e.g., 95% [10]. The latter not only boosts its output voltage to a target level, but also supplies a large amplitude of current,

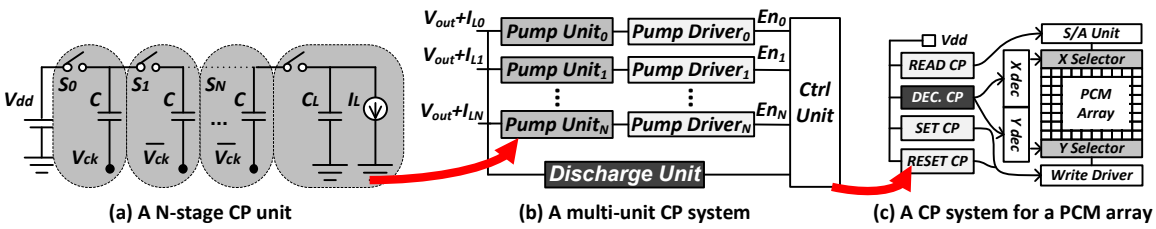

Figure 4: Charge pump basics.

which are essential for READ, RESET and SET operations. It incurs large die area overhead and has low power conversion efficiency, e.g., 20% [21]. Figure 4(c) shows a CP system [21] for a diode-switched PCM chip using both types of CP. The capacitive load pumps are used for X/Y decoders. The READ, SET, and RESET operations require voltages of 3.0V, 3.0V and 5.0V respectively, in contrast to a 1.8V V_{dd}. They require current load pumps which have low efficiency and large die area overhead. When current load CPs step up the voltage for incoming reads or writes, they are effectively capacitive load CPs which have high efficiency. However, when a read or write begins to drain current from the CPs, the efficiency drops drastically because it is a strong decreasing function of the load current [35].

When there is a need for more output current, e.g., writing multiple MLC cells, either a larger CP is needed and/or more modular pump units are needed such that the aggregated current matches the demand for writing multiple cells. In this paper, we choose to integrate modular CP units for their flexibility in adjusting voltage output, as with recent chip demonstrations [21, 17, 6]. The organization of a write CP can be viewed in Figure 4(b). When different numbers of pump units are enabled (by the En_i signal), the circuit can support $\times 2$, $\times 4$, $\times 8$ and $\times 16$ concurrent cell writes. In [21], one pump unit can RESET two cells. We share the same assumption in this paper. Finally, a discharge unit is also necessary. It is responsible for discharging pump units to V_{dd}. The discharge process is an atomic operation that cannot be interrupted.

3.2. Charge Pump Modeling

A CP with a current load is both a power supplier and consumer. Besides the power delivered to the output network, part of the input power is consumed on parasitic effect, and part of the input is leaked away. Also, the peripheral circuits that support the CP core circuits drain considerable power. The *parasitic power* is the power consumed on charging/discharging the internal parasitic capacitance that does not contribute to the output [26, 35], and is proportional to the capacitance of internal pumping capacitors [26]. The parasitic power is a dominant factor in the wasted power of a CP since the internal capacitors are very large [26]. The *leakage power* is the power leaked from supply to ground [35]. Leakage is also quite large as a result of very strong transistors and high voltages on output and internal nodes. The *peripheral power* is the power consumed on supportive circuits such as controls, drivers, clock distribution, etc.

We follow the latest previous work [26] on CP modeling. Since the model does not consider transistor leakage, we also built CPs for different operations in HSPICE and measured the leakage power which is then integrated into the analytical CP model. The HSPICE implementation with design parameters was also used to verify critical metrics against the model. The critical metrics are area, wasted power, charge and discharge latency, and charge energy. All metrics are functions of N, the number of stages used in a pump unit (Figure 4 (a)). We will summarize their analytic models respectively and then provide a design space exploration on those metrics for an optimal selection of N.

Total current supply. The total current is modeled as:

$$\mathsf{TC} = \left[(\mathsf{N}+1) + \alpha \cdot \frac{\mathsf{N}^2}{(\mathsf{N}+1)\cdot \mathsf{V_{dd}} - \mathsf{V_{out}}} \cdot \mathsf{V_{dd}}\right] \cdot \mathsf{I_L} \tag{1}$$

where, α is the constant proportional factor between the bottom plate parasitic capacitance and the pumping capacitance. $\mathsf{I_L}$ is the load current, which is all the current that drains from the CP output that can cause the attenuation of CP output voltage. $\mathsf{I_L}$ mainly consists of three components: 1) the dynamic current used by the load, i.e., read/write current applied to PCM cells, denoted as $\mathsf{I_{L-dynamic}}$. This is the current for doing useful work; 2) the leakage of the load, denoted as $\mathsf{I_{L-output-leak}}$; and 3) the leakage of the CP itself, denoted as $\mathsf{I_{L-CP-leak}}$. Hence,

$$\mathsf{I_L} = \mathsf{I_{L-dynamic}} + \mathsf{I_{L-output-leak}} + \mathsf{I_{L-CP-leak}} \tag{2}$$

Wasted power. The wasted power is calculated as:

$$\mathsf{WP} = \mathsf{TC} \times \mathsf{V_{dd}} - \mathsf{V_{out}} \times \mathsf{I_{L-dynamic}} \tag{3}$$

which includes parasitic power and the leakage power from both the output load and the internal CP circuit.

Area. The die area of a CP is proportional to the maximum current it can provide [26]:

$$\mathsf{A_{tot}} = \mathsf{k} \cdot \frac{\mathsf{N}^2}{(\mathsf{N}+1)\cdot \mathsf{V_{dd}} - \mathsf{V_{out}}} \frac{\mathsf{I_L}}{\mathsf{f}} \tag{4}$$

where $\mathsf{A_{tot}}$ is the total area of the CP, k is a constant that depends on the process used to implement the capacitors, f denotes the working frequency of the CP, $\mathsf{V_{dd}}$ is the supply voltage, $\mathsf{V_{out}}$ is the target programming voltage, and N is the number of stages in a pump unit as shown in Figure 4 (a).

Charge/discharge latency. The charge and discharge latencies indicate the time spent in raising the output voltage to the target voltage level, and from the target voltage to V_{dd}, respectively. For the same CP configuration, the higher the

output voltage is, the longer time the CP spends in charging/discharging.The charge latency, t_r, is calculated as:

$$t_r = T \cdot N^2 \cdot \left(\frac{C_L}{C_{Tot}} + \frac{1}{3}\right) \cdot \ln\left(\frac{N+1-v_{x0}}{N+1-v_x}\right) \quad (5)$$

where T is $\frac{1}{f}$, the period of a CP. C_L represents the total capacitance load. C_{Tot} means the total pumping capacitance. v_{x0} is initial voltage and v_x is target voltage.

Charge energy. The charge energy is the energy consumed during CP rise time. And it is defined as:

$$Q(t_r) = (N+1)\left(\frac{1}{3}C_{Tot} + C_L\right)(v_x - v_{x0}) + \alpha C_{Tot} V_{dd} \frac{t_r}{T} \quad (6)$$

With all parameters considered, the model details of CPs are summarized in Table 4 in Section 6.

From equation (3)-(6), we can see that all essential metrics are functions of N. To achieve an optimized design, we performed a design space exploration on N. Figure 5 exhibits the results for RESET and READ CPs in our baseline. The SET CP design follows the READ CP as they have the same output voltage. We normalize all values to the minimal value of the corresponding metric for ease of comparison. For the RESET operation, We select N=3 as the stage number for the RESET CP for its low overhead in all metrics including wasted power (Wasted), charging latency (Tcharge) and energy (Echarge). This setting will be used in our baseline design, which will be compared against our proposed technique that uses a hybrid design with two different stage numbers for overall power and energy savings. For the READ CP, the smallest overhead and the best performance metrics are obtained, when the stage number is one (N=1). Therefore, we adopt a single stage CP design for both READ and SET operations.

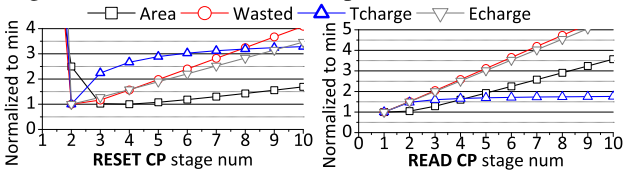

Figure 5: The RESET and READ CP modeling.

Verification and leakage power. We implemented an actual CP design [26, 35] in HSPICE with target requirements and the optimal stage number calculated above, to verify the metrics against the model. The CPs are implemented using high voltage transistors and MOS-capacitors. We measured the leakage power and summarized them in Table 1. The leakage power of each specific CP is proportional to its area and stage number, since more stages incur a larger number of transistors. Compared to our HSPICE simulation results, the metrics calculated from equation (3)-(6) have less than 10% deviation. We used values obtained from the model in our designs and evaluations.

CP Type	RESET	SET	READ
Leakage Power (mW)	2.0	0.42	0.51

Table 1: Leakage power for all types of CPs.

CP reliability under high output voltage. The reliability is an important issue for CP design. Previous works [18, 19, 26] demonstrated the occurrence of gate-oxide overstress in CPs, which leads to severe threat to reliability. In recent chip demonstrations, all CPs discharge their high output voltage forcefully to V_{dd} [17] during idle period when there is no request. The active discharging operations lead to reduction of standby currents. More importantly, it maintains the device reliability of a CP by minimizing its duration under high voltage exposure [17]. The transistor gate oxide of CPs is particularly vulnerable under long application of high voltage. The breakdown is caused by formation of a conducting path through the gate oxide to substrate due to electron tunneling current. This is the so called time-dependent gate oxide breakdown (TDDB) [13]. The relationship between voltage and TDDB duration can be calculated by [4]:

$$T_{DB} = A \cdot \exp(-\gamma \cdot \sqrt{V}) \quad (7)$$

where, T_{DB} is time to failure. A and γ are constant. And V denotes the operating voltage. As we can see, T_{DB} is inversely exponentially proportional to the root of voltage, so a slight growth of voltage results in significant degradation on time to TDDB failure.

A CP could be built such that its transistors are much less high-voltage stressed. Unfortunately, such CPs have much more stages than our baseline design, since it will have much larger area, lower power efficiency, and higher latency. This was investigated by the design space exploration shown in Figure 5. As we will show in Section 4.1, a CP with reliability management will improve its lifetime by 9×. Otherwise, 9× more CPs would have been adopted to achieve the same reliability, which is impractical giving that the current CPs already occupy ∼30% of the chip area (Figure 1(b)). Therefore, we chose the design that yields the best area/leakage/latency, and manage its reliability dynamically.

4. Proposed Designs

4.1. Motivation

Besides the wasted power dissipated in CPs, the memory interface, peripheral circuit of the data arrays also dissipate significant leakage power. Previous work proposed to power gate peripheral circuits upon completion of an operation with small performance penalty [38]. As we will show next that the wasted power in CP predominates the leakage dissipated in the peripheral circuits of data arrays, especially when large CPs are used for high-voltage operations such as RESET.

Figure 6 shows a power breakdown of a diode-switched PCM chip (22nm 8GB 2-bit MLC) in our baseline. We show both dynamic and wasted power (mainly parasitic and leakage power) for the RESET CP, SET CP, READ CP, and all remainder components (Other) including the data bank itself, peripheral circuits and the LPDDR2 interface. The CP for X/Y decoders has

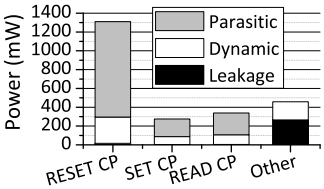

Figure 6: Power breakdown.

only capacitive load and hence dissipates negligible power, compared to other components. To obtain those results, we modeled the PCM chip architecture following mainly a recent prototype [6]. The bank itself was modeled using NVsim [8]. The LPDDR2 modeling is based on [23, 22]. More experimental parameters can be found in Section 5.

It is clear from the results that the bulk of power dissipation goes to all CPs, especially the RESET CP, due to their large target output voltages, but low power conversion efficiency. The V_{dd} of the PCM bank is 1.8V [6], and the LPDDR2 interface has 1.2V V_{dd} [22], but the working voltages for READ, SET and RESET are 3.0V, 3.0V and 5.0V respectively. Also, only around 20% of conversion efficiency (Dynamic power / (Dynamic power + Wasted power) $\approx$ 20%) was observed for RESET CP, similar to that reported in [21]. As a result, most of the power is wasted on CPs. The larger the CP is, the more wasted power it has.

Moreover, the total power dissipated by RESET CPs is more than 50% of the total PCM chip power. This is not only because the RESET by itself has the highest power requirement, but also because its CP is designed to accommodate the worst case scenario where there are multiple concurrent RESETs. More pump units are required to satisfy such peak power demand. And hence, a larger area is required and more power waste is generated. If the peak power demand is reduced, we can then use smaller CPs to achieve both area and wasted power reduction. This is one objective of our design in this work.

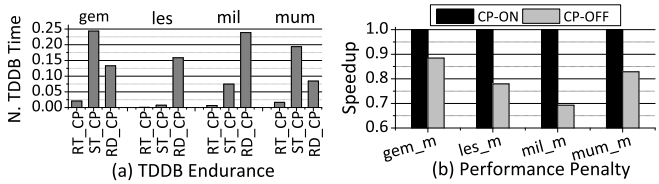

Figure 7: TDDB endurance and performance.

Next, in order to understand the importance of the reliability of the CPs under a per-operation switching (POS) scheme, we compared its TDDB against the scheme that always keeps CP on. Figure 7(a) shows the TDDBs of the latter scheme, normalized to POS, for RESET (RT_CP), SET (ST_CP), and READ (RD_CP) CPs respectively. As we can see, the lifetimes of CPs in general degrade to less than 25% of POS, which proves the necessity of per-operation switching. However, POS will inevitably experience large performance degradation, as evaluated in Figure 7(b) (experimental setting in Section 5). "CP-off" means that CPs are turned off as soon as an operation is finished, and turned back on when a new request arrives. When CPs are turned off, the LPDDR2 interface, peripheral circuits and data banks are all turned off. LPDDR2 interface has a short wakeup time as it does not have on-die termination and delay locked loop, and is considered as the most power efficient interface [22]. Peripheral circuits and data banks do not introduce significant power-on latency [38]. CPs, on the other hand, have long charge and discharge latencies that override other power-on overheads. In the figure, "CP-off" is normalized to "CP-on", and the performance loss is between 6.4% to 15%. Such degradation would increase the overall energy consumption as the applications now run slower. Hence, our second objective is to develop a fast power-on scheme of CPs and combat such significant performance loss.

Our solution. In this paper, we focus our design on optimizing the uses of CPs via a two-step approach. We first reduce the peak power requirement of an MLC PCM so that the RESET CP can be shrunk significantly. Then, we reduce the power-on time of performance critical CPs. The first step reduces the magnitude of the wasted power in RESET CPs while the second step minimizes performances loss. We will elaborate the design details of our two-step approach in the following sections.

4.2. Step I: Intra-write RESET Scheduling (Reset_Sch)

The P&V programming strategy for writing an MLC PCM line starts with a RESET iteration, followed by a non-deterministic number of SET iterations. In particular, the number of SET iterations is cell value dependent [16]. We use a 2-bit MLC as an example for illustration in this paper. On average, writing value "01" requires more iterations than writing "10" and "11", while writing "00" can finish immediately after the RESET iteration. A RESET consumes 3.3$\times$ the power of a SET, but is twice as fast. When writing a PCM line, all changed cells start with the high-power RESET, generating the largest power draw from the RESET CP. Hence, the power consumption reaches its peak in the first iteration and drops dramatically in the following iterations. Figure 8 illustrates this scenario. The size of a CP is determined by the power demand of its load. The higher peak power one write has, the larger CP it requires. Hence, it is crucial to reduce the peak power of a write in order to shrink the size of a CP.

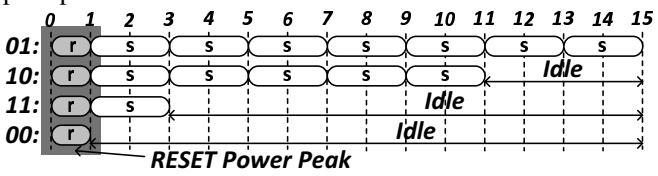

Figure 8: The power requested to write a PCM line reaches its peak in time slot 0 (horizontal numbers indicate time slots, "r"/"s" stands for RESET/SET).

To prevent the power peaks from concurrent RESETs, we observe that the latency of a line write is determined by the slowest cells, typically those writing a "01". Other cells only need to complete no later than those slowest cells to maintain performance. This implies that we can defer the RESETs of faster cells as long as their SET iterations can finish with or before the slowest cells. In other words, we can schedule and minimize the concurrency of the RESETs without lengthening the write latency. Hence, the peak power is distributed across the entire write procedure, greatly reducing the pressure on the CP. We can use smaller CPs to satisfy all

RESETs within a write without increasing its latency. In addition, scheduling RESETs can be applied to single level cell (SLC) PCM as well. Since RESET is slightly more than twice as fast as a SET, we can split all RESETs of a line into two groups and finish them within the duration of a SET. Evaluation on such opportunity will also be given in Section 6.

However, exceptions can occur since MLC PCM write exhibits significant non-determinism [2, 24], and writing a "10" may be slower than writing a "01" occasionally, which prolongs the write latency if the RESET of the former is delayed. Nevertheless, scheduling RESETs according to the *average number* of SET iterations for different cell values imposes a negligible impact on write latency, which will be shown in our experiments and results.

The RESET scheduling (Reset_Sch) mechanism we propose is a simple heuristic based on the average number of SET iterations for writing four different values, as depicted in Figure 9. Let `S-slack` denote the time slot interval in which a RESET iteration can be scheduled without prolonging the write latency, assuming there is always a cell needed to be changed to "01". Hence, the `S-slack` for writing "10" is 4, "11" is 12 and "00" is 14, as depicted in Figure 9. Note that the `S-slack` for "00" is aligned with "11" in the figure because 1) it simplifies the scheme; 2) it does not bring much benefit otherwise, according to our experiments; and 3) occasionally a line may take less time than predicted so that finishing "00" sooner actually benefits. The scheduling works as the following:

1) For "01" cells, we do not defer their RESETs.
2) For "10" cells, we defer their RESETs until after all RESETs for 1) have finished, but before the end of their `S-slack`, which is 4 in Figure 9.
3) For "11" and "00"cells, we defer their RESETs until after all RESETs for 2) have finished, but before time slot 12 in Figure 9.

In all above cases (and baseline design as well), the concurrency degree of RESETs is bounded by the capacity of the RESET `CP`. For example, if the RESET `CP` can support *R* RESETs per time slot but there are more than *R* RESETs, we then spill the extra RESETs to the next time slot, and so on. In some extreme cases, such spilling could go beyond the `S-slack`, postponing all subsequent cell writes and prolonging the latency of writing the entire line. Such scenario can be mitigated through value encoding (as described below), and proper selection of *R*, as will be demonstrated in our evaluations and results.

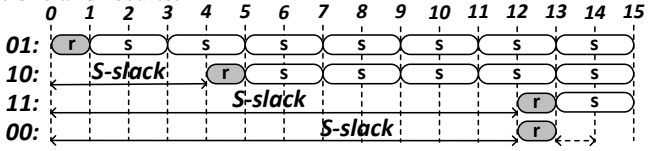

Figure 9: Scheduling RESETs without harming write latency.

Value encoding. Our Reset_Sch favors to write "11" and "00" as they have larger `S-slack`s than "10" and "01". We adopt a recently proposed data value mapping scheme [34] to create more "11"s and "00"s for a memory line. This encoding scheme divides cell values into two categories: "11"/"00", and "01"/"10", and dynamically remaps values to ensure a line contains more cells in the first category. Since the cells in the first category consume less per-cell write energy, the new mapping reduces dynamic write energy of MLC PCM. We further extend this encoding by flipping the bits if there are more "01"s than "10"s in a line. The hardware overhead involves only a two-bit MLC cell tag per line. A read operation, before returning the line, interprets this tag and restores the original values if necessary. The hardware cost is trivial as shown in [34]. When combined with differential write [38, 20], the remapping is disabled if there are more cell changes, the same as that in [34].

Comparing to prior art. The schemes that are close to Reset_Sch include T_{off} *scheduling* [12] and *Multi-RESET* [15]. Both schemes manage SET and RESET iterations within one line write.

- T_{off} scheduling. For P&V MLC programming, a cell being changed needs a delay to stabilize resistance drift and recover V_{th} between each SET and read/verify pair. This delay is referred to as T_{off} in [12]. A T_{off} skew write scheme was proposed to interleave SETs from multiple cells, which maximizes SET throughput of MLC PCM without increasing SET current or SET pump size. This scheme has no impact on RESET pump size because RESET operation does not have T_{off}, and scheduling SETs does not conflict with scheduling RESET.
- Multi-RESET. To enable more MLC line writes with fixed input power budget, the Multi-RESET scheme proposed in [15] divides all RESETs into several groups and each group can be performed with currently available power budget. Multi-RESET, while starting MLC write early, prolongs the latency of this particular MLC line write. Therefore, performing Multi-RESET too aggressively would harm system performance [15]. We will compare Reset_Sch with Multi-RESET in the experiments.

4.3. Step II: Charge Pump Scheduling (CP_Sch)

The per-operation switching (POS) scheme of `CP`s is most beneficial to maintaining their reliability. In addition, since memory idle time predominates over busy time, POS also presents much opportunity for leakage energy saving. However, charging the `CP`s consumes extra dynamic energy (the energy that was discharged). It should not become an overkill to the leakage saving. It turned out that such dynamic energy overhead can be offset by the leakage savings during idle time, as will be shown in Figure 18. That is, POS will not incur additional energy overhead, but will generate significant performance degradation, as shown earlier in Figure 7(c). In this section, we study how `CP`s should be designed and managed to have minimal impact on performance.

4.3.1. Write `CP` Management. The POS scheme for writes is relatively straightforward. This is because writes are typically

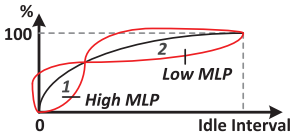

Figure 10: CDF of the idle interval.

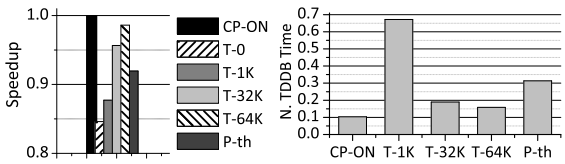

Figure 11: Tradeoff between performance and the reliability of READ `CP`. T-x means the `threshold` is x cycles.

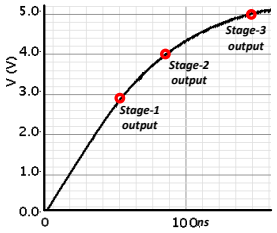

Figure 12: 3-stage RESET `CP` charging process.

issued to the memory banks in a bursty fashion. This pattern indicates that the on time for RESET `CP` is quite predictable. The charge and discharge happen right before and after a RESET. The bursty writes can be leveraged to hide the charging latency of the next write with the previous write. For the SET `CP`, the P&V write scheme ensures that the `CP` remains on for a long time to perform a sequence of SET and verify operations, following an initial RESET. Hence, with the bursty write pattern, the SET `CP` charges at the beginning of a burst and discharges only at the end of a burst. The intermediate RESETs are too short to switch the SET `CP`. Our experiments also indicate that the charging time for write `CP`s has insignificant impact on performance. The READ `CP`, however, does because reads are performance critical and they are not bursty.

4.3.2. Read `CP` Management. Our analysis on the reliability of a `CP` indicates that the longer idle time a `CP` can exploit (i.e. stays off), the longer its lifetime can be. This is consistent with POS, but harmful from the performance standpoint, as we have shown earlier. Hence, it is necessary to study this tradeoff and develop scheme to achieve the best performance with the minimum impact on reliability.

Characterizing READ `CP` idle time. Unlike writes in MLC PCM, reads are single-operation requests and they are not as bursty as writes. Hence, when to turn on and off the `CP` depends on how reads are spaced apart. Previous work on leakage reduction for DRAM have also exploited rank idleness to power down memory ranks [14, 23]. The basic approach is to keep memory rank on, after serving a request, for a `threshold` amount of time anticipating future memory requests. If no request arrives, one memory rank shifts to a deep power-down mode to save leakage energy. The threshold value can also be dynamically adjusted based on prediction of the request arrival time [36]. While a threshold based scheme can be adopted here, we describe the major challenges of such a scheme and introduce our new `CP` design to overcome the limitations.

A threshold based shutdown scheme maintains performance for those reads that arrive sooner than the `threshold`, but benefits reliability for the duration beyond the `threshold` till the next request arrives. The charging delay of the READ `CP` for the next read then becomes a performance penalty since it is difficult to predict the arrival time of the next read and leave headroom for charging. We now characterize the relation among reliability, performance impact and the value of the `threshold`. Figure 10 sketches the CDF of the idle intervals between consecutive memory requests. When idle intervals are relatively small (i.e., region 1 in the figure), memory requests are densely spaced, indicating that the memory level parallelism (MLP) is high. Inserting charging overhead to the requests in this region is likely to have small influence on the overall system performance since the overhead can be well hidden. On the contrary, when idle intervals are in region 2, memory requests are more isolated, indicating a low MLP. Hence, it is more difficult to hide any charging overhead in this region as requests are more sequential [31]. As a result, if a `threshold` is in region 1, the requests in region 2 bear the charging overhead, which results in significant performance loss. On the other hand, when the `threshold` is in region 1, better reliability can be achieved as the off time of the `CP` is relatively long. In summary, a `threshold` in region 1 incurs better reliability but more performance loss. A `threshold` in region 2 incurs less reliability but less performance loss. This intuition is quantified in Figure 11 where the performance (left), normalized to `CP`-always-on (`CP-ON`), and TDDB (right), normalized to POS, are compared when a `threshold` varies from 1K (`T-1K`) to 64K (`T-64K`) cycles. Note that `T-0` is POS itself. As expected, the performance improves but the reliability degrades with larger `threshold` values. We also show the adaptive threshold scheme based on last value prediction of the idle interval (`P-th`) [36]. `P-th` achieves only moderate performance and reliability since there is little correlation between the next idle interval and the history, and so the prediction accuracy cannot be guaranteed.

Instead of following the tradeoff study for `threshold` selection, we decide to choose a small `threshold`, e.g., 0 or 1K cycles, since they offer much better reliability. To tackle the low performance, we design a new pumping technique to provide a fast charging-up for incoming read requests.

A new charging scheme for reads. The charging process of a `CP` follows a non-linear voltage rising curve. A `CP` arrives at an intermediate voltage level very quickly, and then slows down when approaching the target voltage. The output voltage is charged up stage-by-stage. The intermediate voltage levels produced by middle stages are achieved much faster than the target voltage supplied by the last stage. Figure 12 shows the charging process of a 3-stage RESET `CP`, simulated in HSPICE. The output of three intermediate stages are

marked on the curve. This figure shows that stage 1 and 2 rise much faster than stage 3. This observation enlightens us to utilize the middle stages of a CP to produce the working voltage of a read. Since such a CP must output a higher target voltage by itself, only the RESET CP can be used for this purpose. Hence, we build an extra RESET CP as an alternative to the READ CP for a fast charge-up operation.

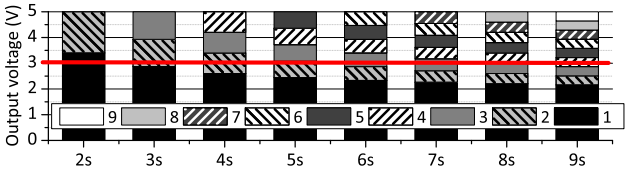

Figure 13: Output voltage of each stage in RESET CP.

However, not all intermediate stages of a RESET CP output an exact 3.0V read voltage. To find out what kind of RESET CP can satisfy this requirement, we tested different sizes of RESET CP with increasing number of stages, and enumerate the output voltage of all their intermediate stages, as shown in Figure 13. Only those CPs that have an stage, at the end of which outputs an exact 3.0V can be used as an alternative to a READ CP. As we can see, a 5-stage RESET CP can produce 3.0V at the end of stage 2, and an 8-stage CP can do the same at the end of stage 3. Hence, these two CPs are suitable candidates for fast READ CPs.

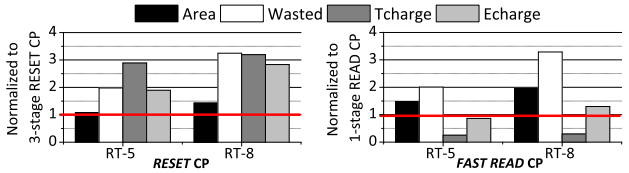

Figure 14: Using the RESET CP as a fast READ CP.

The next question is: which one is a more economical design for reads between a 5-stage RESET CP and an 8-stage RESET CP? Our baseline 3-stage RESET CP was chosen for a better balance among area, power wastage, charge latency and charge energy. The 5-stage and 8-stage RESET CP have higher measures in all metrics than the 3-stage baseline design, as compared in Figure 14 (left). However, both CPs outperform the baseline READ CP, which uses one stage only, in charge latency, as shown in the figure (right part). The 5-stage design can charge to 3.0V about 4 times faster than the READ CP does, which is very attractive from a performance perspective. Moreover, its charge energy is also less than of the READ CP. For these two reasons, we choose the 5-stage RESET CP as a FAST READ CP for serving only READs. The area of a FAST READ CP is about the same as a 3-stage RESET CP, but 50% more than a 1-stage READ CP. However, as we will show later, the overall area of CPs is still reduced thanks to Reset_Sch. One drawback, however, is that the total power wastage of a FAST READ CP is twice as much as the original READ CP. It is possible to keep both the FAST and original READ CPs, and use the former for charging the first read, and the latter for subsequent bursty reads with the former turned-off. We term this scheme Switch in our following evaluations.

4.4. Summarizing CP Count

We now summarize the total CPs required after applying both Reset_Sch and CP_Sch schemes. Since CPs are of different sizes, we will normalize all area to the baseline 3-stage RESET CP for ease of comparison. Our experimental results will show that the number of RESET CP units can be reduced to only 30% of the baseline without performance degradation. Additional 5-stage RESET CPs are necessary to serve as the FAST READ CP, each is 1.07× the area of the baseline 3-stage RESET CP. One READ requires 8.4μA for each cell. One chip should support 512 cell READs, requiring 4301μA. One FAST READ CP supplies $3V$ and $100\mu A$. Hence, we need ∼43 FAST READ CPs which amounts to the area of ∼46 3-stage RESET CPs. The number of SET CP remains the same. The final counts are listed in Table 2. As we can see, the total area of all CPs is reduced by 3% even though the FAST READ CPs are larger.

	RESET	READ	SET	Total
Before	70	13	11	94
After	21	59	11	91

Table 2: The RESET CP count.

4.5. Hardware Overhead

We synthesized the control logics of Reset_Sch and CP_Sch by Synopsys design compiler and IC compiler. The total area overhead at 45nm is 251.34μm^2, which is similar to the area of 3KB PCM cells. Reset_Sch spends 0.9ns and 1.18pJ in one scheduling operation. CP_Sch costs 0.41ns and 0.91pJ per operation.

5. Experimental Methodology

Simulator: We evaluated our proposed designs and techniques using a PIN-based simulator Sniper [3]. We modified the simulator to model all memory hierarchies, power budgeting constraints [15], and CP system.

Baseline configuration: The detailed baseline parameters can be found in Table 3. Unless otherwise stated, we adopt diode-swtiched MLC PCM based main memory system. Other types of PCM main memory results are also reported.

Main memory configuration: Our memory controller prioritizes read requests. Write requests are only scheduled when there is no read request. When the write queue is full, the memory controller issues a write burst, where all pending read requests are blocked until all writes in the queue are finished [11, 15]. Write scheduling must obey not only bus and chip scheduling constraints, but also the local and global power budgets supplied by CPs.We adopted both local and global CPs [15] in our baseline.

We considered a main memory with two 8GB MLC PCM LPDDR2-NVM channels [6]. One DIMM has one channel, two ranks and four banks per rank. A bank spreads across four 16-bit wide chips. Therefore, in each rank, four banks

share four PCM chips.

Chip modeling: We used NVsim [8], a CACTI-based non-volatile memory modeling tool, to calculate our 22nm diode-switch PCM chip parameters. The chip modeling is based on the latest industrial prototype [6]. We modeled CP as discussed in Section 3.2, fitting constants in [26] and related parameters in [6, 21]. We calculated read and write latency/energy by feeding the parameters from [6, 21] to NVsim. We used the same non-deterministic write model as previous work [30, 15]. The detailed memory, chip, CP, read and write parameters can be found in Table 4.

Simulated workloads: We chose a subset of programs from SPEC2006, BioBench, and STREAM suites to construct multi-programmed workloads covering different memory access characteristics. The workload mix_1 consists of astar, bwaves, gemFDTD and leslie3d, and mix_2 consists of astar, bwaves, milc and mummer.

Simulation and evaluation: The representative phases of benchmark was chosen using PinPlay [28]. We simulated 5 billion instructions to obtain performance results. For our results, we define *speedup* as: $\text{Speedup} = \frac{\text{CPI}_{\text{baseline}}}{\text{CPI}_{\text{tech}}}$, where $\text{CPI}_{\text{baseline}}$ and CPI_{tech} are the CPIs of the baseline setting and the setting with scheme tech, respectively. This metric is used in related research [16, 30].

6. Results and Analysis

6.1. Reset_Sch: Power Reduction and Performance

We will present first how much area reduction for RESET CPs is achieved, and then show the impact of such reduction on performance. The wasted power reduction is proportional to the area reduction. Since we adopt a modular CP design, its area is thus linear to the number of modular pump units which decides the peak power provisioning to the memory, or how many RESETs can concurrently occur. This is the capacity of the RESET CP, and a critical parameter in our design. A small capacity implies small CP size but spilling of RESETs to the next iteration may occur in each step of the scheduling, creating performance degradation. A generous capacity implies a large CP size, but more RESETs can be packed in each iteration and writing a line can finish sooner. Our objective is to identify the smallest RESET CP that does not bring performance overhead. We compare the following three choices of the capacity with the baseline.

- Base: Our baseline implements the state-of-the-art fine-grained power budgeting technique including the multi-RESET scheme [15], which makes the best use of available power for higher performance. All RESET CPs are on in Base with power gating on arrays during idle periods [38].
- SumofMax: This builds Reset_Sch on top of Base. The capacity of the RESET CP supports ($max_{01} + max_{10}$) number of concurrent RESETs, where max_{xx} is the maximal number of *xx* occurrence in writing a single line throughout the simulation duration. This scheme guarantees that writes are not prolonged since there is enough power to write "01" and "10" at the same time.
- Max_{01}: The capacity is now reduced to max_{01}, presuming $max_{01} > max_{10}$ as writing "01" is the longest operation.
- MinofMax: The capacity is reduced further to min(max_{01}, max_{10}). This scheme utilizes our enhanced version of value encoding to reduce max_{01} through bit flipping if there are more "01"s then "10"s in a line.

Figure 15 reports the required sizes of the RESET CP for three different capacities, normalized to the Base. This figure also reflects achieved wasted power reduction for RESET CPs. The results show that SumofMax only requires 70% of the RESET CPs in Base. Max_{01} and MinofMax can further shrink the sizes down to 38% and 33% of the RESET CPs in Base. As we can see, opportunities in reducing the wasted power and area of a RESET CP through Reset_Sch are substantial. This does not have to be done at the cost of performance, as we show next.

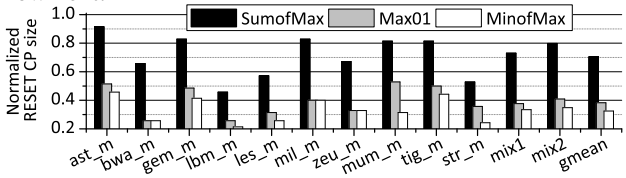

Figure 15: The potential of CP size reduction and wasted power reduction.

Figure 16 compares the performance of SumofMax, MinofMax and MinofMax-30% (explained later), all normalized to the Base. It is interesting to observe that SumofMax and MinofMax improvement performance over Base by 2.6% and 2.1% respectively. The reason is the following. The RESET CP in Base is configured from empirical studies on peak power requirement. However, when bursty writes arrive at the memory, often there are RESETs in the first iteration that cannot be served due to insufficient power. And they are spilled to the next iteration, a.k.a. multi-RESET [15], which prolongs the write latency. With Reset_Sch, however, such spilling occurs much less often *even when the* CP *size is greatly reduced*. Hence, we can achieve slight performance gains while reducing the cost of RESET CP. When we use a fixed CP size for all benchmarks, e.g., use 30% of the RESET CP in Base as denoted by MinofMax-30%, the average performance improvement drops to 1.5%. Our experiments indicate that below this size, the performance gain becomes negative. Hence, we select the size of MinofMax-30% in the following experiments.

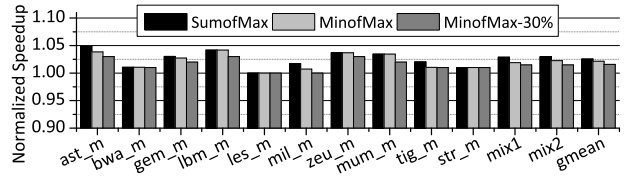

Figure 16: Performance evaluation with reduced CP sizes.

6.2. CP_Sch: Performance, Energy and Reliability

As we mentioned earlier, the naïve POS scheme bears noticeable performance overhead. This is shown in Figure 17 as

CPU	four 4GHz, X86 out-of-order cores, 4-wide issue, 8MSHRs/core, 128-entry instruction window
L1 I/D	private, I/D 32KB each/core, 4-way, LRU, 64B line, 2-cycle hit
L2	private, 2MB/core, 8-way, LRU, 64B line, write back, 2-cycle tag, 5-cycle data hit, 16-cycle CPU to L2
DRAM L3	private, offchip, 32MB/core, 16-way, LRU, write back, 64B line, 50ns (200-cycle hit), 64-cycle CPU to L3
Memory Controller	onchip, 24-entry R/W queues, MC to bank 64-cycle, scheduling reads first, issuing writes when there is NO read, when W queue is full, issuing write burst (only scheduling writes and delaying reads, when W queue is empty)
Main Memory	16GB, 64B line, 2 channels, 1 channel per DIMM, 2 ranks per channel, 4 banks per rank, 64-bit channel width

Table 3: Baseline configuration

Memory	22nm PCM process, LPDDR2-NVM interface, I/O bandwidth: 800Mb/s/pin, 4 chips per rank, 4 banks interleaved on 4 chips
Chip	4 F^2 cell, diode-switch, 16-bit width, 1GB, 1.8V vdd, 133MHz, 140 concurrent RESETs power budget
CP per Chip	133MHz, RESET/SET/READ CP: 5/3/3V target voltage, 7/3.5/4.3mA load current, 21.6%/30.9%/30.9% power efficiency 159/132/132ns charge latency, 100/87.5/87.5ns discharge latency, charge energy 18.78/2.76/3.31/2.86(FAST RD)nJ
Read	3V, 8.4μA, 5.6nJ per line, critical word 125ns, MLC read 250ns
Write	RESET: 5V, 100μA, 29.7pJ per bit, 50ns operation latency, 125ns iteration latency; SET: 3V, 50μA, 22.5pJ per bit 150ns operation latency, 250ns iteration latency (including T_{off} [12]& verify), MLC Write Model: 2-bit MLC [30, 16]

Table 4: 2-bit MLC PCM chip and main memory configuration

T-0 which has an average of 16% slowdown. However, its total energy consumption is approximately the same as Base, as shown in the left of Figure 18, since the charging/discharging energy is offset by the leakage savings upon idle. With FAST READ CP with a threshold (F-threshold), the performance degradation is reduced to 4% and 2% for F-0 and F-1K respectively, thanks to the new CP design that is 4× faster. On the other hand, Figure 18 also shows that FAST READ CP has more parasitic energy (F-0) as was studied in Figure 14. This is then mitigated by the Switch scheme which has identical performance as F-threshold because both use FAST READ CP for an arriving read, but with comparable parasitic power (1.4% higher) to Base and T-0 because the original READ CP kicks in once it is up.

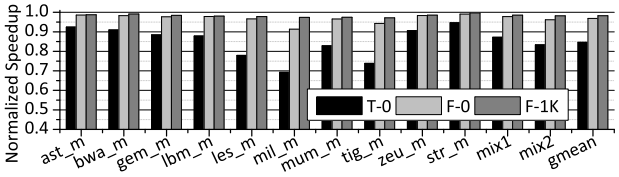

Figure 17: Performance benefit of FAST READ CP.

The reliability of the corresponding schemes are compared on the right part of Figure 18. All results are normalized to T-0. As we can see, Base has the worst lifetime as it always keeps CPs on. F-0 and T-0 are comparable since both actively discharge CPs. Switch, on the other hand, can improve the lifetime of the READ CP by 16% because the on time of both the FAST and original READ CPs are less than T-0/F-0 due to the switch between them.

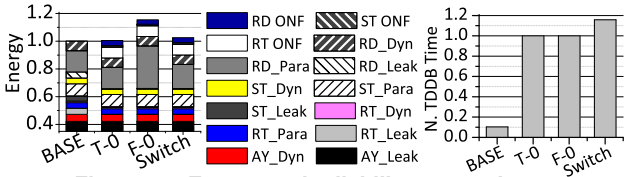

Figure 18: Energy and reliability comparisons.

6.3. Overall System Performance, Power and ED^2

Combining both Reset_Sch and CP_Sch, we now summarize the overall power, total energy, performance, and ED^2 for the whole system including cores and all memory hierarchy in Figure 19. Recent work [23, 29] found that the memory system usually dissipates 25% of power of the entire system. Therefore, in Figure 19, cores and last level DRAM cache occupy 75% of the total system power. Our Reset_Sch reduces the system power to 88% of Base. T-0 increases energy quite a lot over Base because of its large performance loss. Hence, it has the highest ED^2 (62.2% increases). Switch increases total energy by 2.1% over Base and decreases the performance by 2%. Its ED^2 increases by 6.3% over Base. In summary, Switch is as energy efficient as Base, but presents the best reliability of all CPs among Base and the naïve POS scheme (T-0).

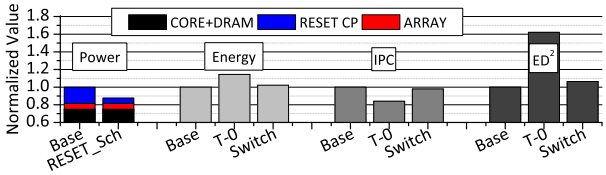

Figure 19: Power, Energy, Performance, and ED^2.

6.4. Extending to other types of PCM

Our proposed Reset_Sch and CP_Sch are not restricted to diode-switched MLC PCM only. They can very well be applied to MOS- or BJT-switched PCM, or SLC PCM. We summarize both power and energy reductions for them in this section. When Reset_Sch is applied, we obtained a power reduction of 49.4%, 37.3% and 40.1% for diode-, MOS- and BJT-switch PCMs respectively. Reductions are smaller for MOS- and BJT-switched PCM because the RESET CP power has smaller proportion in their total power. More interestingly, Reset_Sch narrows the power gap between different designs, making high-density implementation more power efficient. When CP_Sch is applied, Switch improves the lifetime of all three kinds of PCM by 9.68×, with energy increase of 2.6%, 1.8% and 0.7% for diode-, MOS- and BJT-switched PCMs respectively. Compared to the T-0, Switch is faster in all three kinds by 16.7%.

Finally, our schemes, especially the Reset_Sch, can be applied to SLC PCM. Given that the RESET pulse has about 1/2

latency of the SET pulse, we can spread RESETs across the duration of the SET, cutting the required RESET CP by half. By reducing the RESET CP size to 64% of the SLC-based baseline, we found that Reset_Sch can reduce 46% wasted power while showing negligible performance degradation.

7. Related Work

PCM has emerged as a candidate for main memory [20, 32, 38]. The high power consumption of PCM has been studied in recent works. *Differential-write* was proposed [38] to only update cells that store different values. By eliminating unnecessary cell writes, the power consumption of PCM write can be reduced. Cho *et al.* proposed *Flip-n-Write* [5] to guarantee that the power demand for writing one line is reduced to no more than half of the original. Hay *et al.* proposed to estimate changed bit number of each write in the last level cache and schedule writes without violating DIMM level power budget [11]. Jiang *et al.* proposed to perform power budgeting for MLC PCM at iteration granularity and consider both DIMM and chip level power restrictions [15].

Recent chip demonstration [6] proposed write CP *pre-emphasis* to accelerate the charging operation by providing a group of auxiliary RESET and SET CPs. In order to boost the write pumps faster, more area overhead and leakage for extra pumps have to be paid. However, in this paper, we identified that the READ CPs are more important to system performance. And we rebuilt and reorganized RESET pumps to accelerate the charge-up operation of READ pumps.

8. Conclusion

The PCM technology exhibits excellent scalability and density potentials. However, high-density PCM often requires higher voltages than V_{dd}. While on-chip CPs have been integrated in recent PCM chips to provide these raised voltages, their low power efficiency has become a major design challenge. Also, due to reliability reasons, CPs are switched on and off on per-operation basis, which imposes significant performance degradation. Our proposed solution can first reduce the wasted power of the RESET CP by 70%, and then reduce the performance loss over the per-operation based switching scheme from 16% to 2%, while improving the reliability of CPs by 16%.

Acknowledgement

We thank the anonymous reviewers for their feedback. We also acknowledge the support from PCM@Pitt research group. This work was supported in part by NSF CSR #1012070, NSF CCF #1242657 and NSF CAREER #0747242.

References

[1] F. Bedeschi *et al.*, "A bipolar-selected phase change memory featuring multi-level cell storage," *JSSC*, 2009.

[2] M. Boniardi *et al.*, "Impact of material composition on the write performance of phase-change memory device," in *IMW*, 2010.

[3] T. Carlson *et al.*, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in *SC*, 2011.

[4] F. Chen *et al.*, "Correlation between i-v slope and tddb voltage acceleration for cu/low-k interconnects," in *IITC*, 2009.

[5] S. Cho *et al.*, "Flip-n-write: A simple deterministic technique to improve pram write performance, energy, and endurance," in *MICRO*, 2009.

[6] Y. Choi *et al.*, "A 20nm 1.8v 8gb pram with 40mb/s program bandwidth," in *ISSCC*, 2012.

[7] H. Chung *et al.*, "A 58nm 1.8v 1gb pram with 6.4mb/s program bw," in *ISSCC*, 2011.

[8] X. Dong *et al.*, "Nvsim: A circuit-level performance, energy, and area model for emerging non-volatile memory," *TCAD*, 2012.

[9] K. Fang *et al.*, "Memory architecture for integrating emerging memory technologies," in *PACT*, 2011.

[10] R. Guo, "High efficiency charge pump based dc-dc converter for wide input/output range applications," Ph.D. dissertation, Electrical Engineering, North Carolina State University, 2010.

[11] A. Hay *et al.*, "Preventing pcm banks from seizing too much power," in *MICRO*, 2011.

[12] Y. Hwang *et al.*, "Mlc pram with slc write-speed and robust read scheme," in *VLSIT*, 2010.

[13] C. Ih-Chin *et al.*, "Electrical breakdown in thin gate and tunneling oxides," *JSSC*, vol. 20, no. 1, 1985.

[14] Intel, "Intel xeon processor e3-1200 family datasheet," in *Data sheet*, 2011.

[15] L. Jiang *et al.*, "Fpb: Fine-grained power budgeting to improve write throughput of multi-level cell phase change memory," in *MICRO*, 2012.

[16] L. Jiang *et al.*, "Improving write operations in mlc phase change memory," in *HPCA*, 2012.

[17] S. Kang *et al.*, "A 0.1-um 1.8-v 256-mb phase-change random access memory (pram) with 66-mhz synchronous burst-read operation," *JSSC*, 2007.

[18] M.-D. Ker *et al.*, "Design of charge pump circuit with consideration of gate-oxide reliability in low-voltage cmos processes," *JSSC*, vol. 41, no. 5, 2006.

[19] M.-D. Ker *et al.*, "A new charge pump circuit dealing with gate-oxide reliability issue in low-voltage processes," in *ISCAS*, 2004.

[20] B. C. Lee *et al.*, "Architecting phase change memory as a scalable dram alternative," in *ISCA*, 2009.

[21] K.-J. Lee *et al.*, "A 90nm 1.8v 512mb diode-switch pram with 266mb/s read throughput," in *JSSC*, 2008.

[22] K. T. Malladi *et al.*, "Towards energy-proportional datacenter memory with mobile dram," in *ISCA*, 2012.

[23] K. T. Malladi *et al.*, "Rethinking dram power modes for energy proportionality," in *MICRO*, 2012.

[24] D. Mantegazza *et al.*, "Statistical analysis and modeling of programming and retention in pcm arrays," in *IEDM*, 2007.

[25] T. Nirschl *et al.*, "Write strategies for 2 and 4-bit multi-level phase-change memory," in *IEDM*, 2007.

[26] G. Palumbo *et al.*, "Charge pump circuits: An overview on design strategies and topologies," *IEEE Circuits and Devices Magazine*, 2010.

[27] A. Pantazi *et al.*, "Multilevel phase-change memory modeling and experimental characterization," in *EPCOS*, 2009.

[28] H. Patil *et al.*, "Pinplay: a framework for deterministic replay and reproducible analysis of parallel programs," in *CGO*, 2010.

[29] M. K. Qureshi *et al.*, "Preset: Improving read write performance of phase change memories by exploiting asymmetry in write times," in *ISCA*, 2012.

[30] M. K. Qureshi *et al.*, "Improving read performance of phase change memories via write cancellation and write pausing," in *HPCA*, 2010.

[31] M. K. Qureshi *et al.*, "A case for mlp-aware cache replacement," in *ISCA*, 2006.

[32] M. K. Qureshi *et al.*, "Scalable high performance main memory system using phase-change memory technology," in *ISCA*, 2009.

[33] S. Raoux *et al.*, "Phase-change random access memory: A scalable technology," *IBM J. RES. & DEV.*, 2008.

[34] J. Wang *et al.*, "Energy-efficient multi-level cell phase-change memory system with data encoding," in *ICCD*, 2011.

[35] O. Y. Wong *et al.*, "A comparative study of charge pumping circuits for flash memory applications," *Microelectronics Reliability*, no. 4, pp. 670–687, 2012.

[36] D. Wu *et al.*, "Ramzzz: Rank-aware dram power management with dynamic migrations and demotions," in *SC*, 2012.

[37] W. Zhang *et al.*, "Characterizing and mitigating the impact of process variations on phase change based memory systems," in *MICRO*, 2009.

[38] P. Zhou *et al.*, "A durable and energy efficient main memory using phase change memory technology," in *ISCA*, 2009.